



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/730,190	12/05/2000	Kestutis Patiejunas	MS160309.1	7993
27195	7590	01/23/2006	EXAMINER	
AMIN & TUROCY, LLP 24TH FLOOR, NATIONAL CITY CENTER 1900 EAST NINTH STREET CLEVELAND, OH 44114			ALI, SYED J	
			ART UNIT	PAPER NUMBER
			2195	

DATE MAILED: 01/23/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/730,190	PATIEJUNAS, KESTUTIS	
	Examiner	Art Unit	
	Syed J. Ali	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 November 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-16, 18-31, 33-42 and 44-50 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-16, 18-31, 33-42 and 44-50 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This office action is in response to the amendment filed November 21, 2005. Claims 1-16, 18-31, 33-42, and 44-50 are presented for examination.
2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Objections

3. **Claims 10-13, 19, 25-28, and 37-39 are objected to because of the following informalities:**

- a. In the above identified claims, "wherein" has been deleted but no language has been included to replace the transitional phrase, resulting in errors of form. In other claims that have deleted "wherein," it has been replaced with "where." Examiner recommends the objected claims be similarly amended.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

4. **Claims 1-16, 18-22, and 46-50 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.**
5. As per claims 1, 8, and 46, the claimed "software component" is non-statutory as it is not tangibly embodied, as the "software component" is implemented entirely in software. Although

Art Unit: 2195

the claims have been amended to indicate that the claimed “software component” is “effectuated on a machine”, the claim is still directed to a program *per se*, and is not in a form that is useful, concrete, and tangible. Examiner recommends the claim be amended to indicate that the claimed “software component” is fixed on some sort of tangible media, e.g. “a computer readable medium.”

6. Claims 2-7, 9-16, 18-22, and 47-50 are rejected for at least the same reasons as presented for their parent claims, as they fail to present any limitations that resolve the deficiencies of the claims from which they depend.

Claim Rejections - 35 USC § 103

7. **Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert et al. (USPN 6,687,729) (hereinafter Sievert).**

8. As per claim 1, Sievert teaches the invention as claimed, including a client side HTTP stack software component effectuated on a machine that processes requests, comprising:

at least one completion port object (col. 3 lines 20-32);

a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request (col. 3 lines 20-32); and

a client side state machine associated with the at least one request (col. 3 lines 34-65).

9. Sievert does not specifically limit the disclosure to a software component existing on the “client side.” Rather, Sievert discusses a thread pool in general, where any computer that utilizes a thread pool or implements multi-threading may make use of the method for thread pool

Art Unit: 2195

management, whether it is a client, server, or some other computer. A recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art.

Additionally, the state machine disclosed by Sievert refers to the operation of the work queue, which in turn function with respect to individual threads, i.e. the work queue is the data structure by which individual threads are serviced and perform work. There is no limitation in the claims, either explicit or implicit, that prohibits intervening data structures to aid with the processing of requests.

10. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Jones et al. (USPN 6,003,061) (hereinafter Jones).

11. As per claim 2, Jones teaches the invention as claimed, including the client side HTTP stack implementation of claim 1, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

12. It would have been obvious to one of ordinary skill in the art to combine Sievert and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

13. **Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Okano et al. (USPN 6,725,253) (hereinafter Okano).**

14. As per claim 3, Okano teaches the invention as claimed, including the client side HTTP stack implementation of claim 1, further comprising a DNS thread that resolves domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

15. It would have been obvious to one of ordinary skill in the art to combine Sievert and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

16. **Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia et al. (USPN 6,493,749) (hereinafter Paxhia).**

17. As per claim 4, Paxhia teaches the invention as claimed, including the client side HTTP stack implementation of claim 1, further comprising a timeout thread with a list of active sockets and timers associated with each socket, the timeout thread selectively times-out at least one socket according to at least one timer in the list (col. 41 lines 19-28).

18. It would have been obvious to one of ordinary skill in the art to combine Sievert and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a

Art Unit: 2195

thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

19. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia as applied to claim 4 above, and further in view of Jones.

20. As per claim 5, Jones teaches the invention as claimed, including the client side HTTP stack implementation of claim 4, farther comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

21. It would have been obvious to one of ordinary skill in the art to combine Sievert, Paxhia, and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

22. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia in view of Jones as applied to claim 5 above, and further in view of Okano.

Art Unit: 2195

23. As per claim 6, Okano teaches the invention as claimed, including the client side HTTP stack implementation of claim 5, further comprising a DNS thread that resolves domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

24. It would have been obvious to one of ordinary skill in the art to combine Sievert, Paxhia, Jones, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

25. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia as applied to claim 4 above, and further in view of Okano.

26. As per claim 7, Okano teaches the invention as claimed, including the client side HTTP stack implementation of claim 4, further comprising a DNS thread that resolves domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

27. It would have been obvious to one of ordinary skill in the art to combine Sievert, Paxhia, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

28. Claims 8, 23, 35, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM Technical Disclosure Bulletin ("Control of Dynamic Threads Pool for Concurrent Remote Procedure Calls") (hereinafter IBM).

29. As per claim 8, IBM teaches the invention as claimed, including a machine effectuated software component that implements a client side HTTP stack, comprising:

a thread pool comprising N threads that process M requests from a client application component, where N and M are integers greater than 1 and where M is greater than N (pg. 199); and

a state machine associated with each of the M requests (pg. 199).

30. IBM discusses managing a thread pool for requests made by an application server, without explicitly indicating the thread pool is for use on the "client side". While the remote server may typically handle RPC calls, there is no reason to believe that the thread pool could not be implemented on the client side. A recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim. In this particular case, the thread pool of IBM could be easily implemented on the client side.

31. As per claim 23, IBM teaches the invention as claimed, including a method effectuated at least in part by a machine for implementing a client side HTTP stack, comprising:

processing M requests from a client application component using a thread pool comprising N threads, where M and N are integers greater than 1 and where M is greater than N (pg. 199); and

associating a state machine with each of the M requests (pg. 199).

32. As per claim 35, IBM teaches the invention as claimed, including a computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, where M and N are integers greater than 1 and where M is greater than N (pg. 199), and associating a state machine with at least one of the M requests (pg. 199).

33. As per claim 46, IBM teaches the invention as claimed, including a machine executed software component for implementing a client side HTTP stack, comprising:

means for processing M requests from a client application component using a thread pool comprising N threads, wherei M and N are integers greater than 1 and wherei M is greater than N (pg. 199); and

means for assigning each of the M requests with a state machine (pg. 199).

34. **Claims 9-13, 17-19, 24-28, 32-34, 36-39, 44-45, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert.**

35. As per claim 9, Sievert teaches the invention as claimed, including the software component of claim 8, further comprising at least one thread activation component that activates at least one of the N threads based on an event (col. 3 lines 45-52).

36. It would have been obvious to one of ordinary skill in the art to combine IBM and Sievert since the method of IBM is absent guidance as to how threads are handled in terms of sending

and receiving data. IBM is limited to showing a method for initializing and controlling the size of a thread pool. Sievert provides additional functionality for a pool of threads to handle work requests as well as encapsulating requests and responses within an I/O completion port, thereby easing the manner in which requests are handled. The use of a completion port is beneficial in that it simplifies distributed computing for multiple concurrent requests by handling all incoming and outgoing data.

37. As per claim 10, Sievert teaches the invention as claimed, including the software component of claim 9, where the at least one thread activation component is a completion port (col. 3 lines 20-32).

38. As per claim 11, Sievert teaches the invention as claimed, including the software component of claim 9, where at least one of the N threads deactivates itself and return to the thread pool when an operation being processed by the at least one of the threads is pending (col. 5 lines 26-38).

39. As per claim 12, Sievert teaches the invention as claimed, including the software component of claim 11, where the event is the receipt of a completion packet by the at least one thread activation component (col. 3 lines 27-32).

40. As per claim 13, Sievert teaches the invention as claimed, including the software component of claim 12, where the at least one thread activation component is a completion port (col. 3 lines 20-32).

41. As per claim 18, Sievert teaches the invention as claimed, including the software component of claim 9, further comprising at least one key associated with at least one of the M requests, wherein a first one of the N threads is associated with the at least one of the M requests, and the thread activation component associates the context of the first one of the N threads with the at least one state machine using the at least one key, in order to activate the first one of the N threads (col. 5 line 59 - col. 6 line 54).

42. As per claim 19, Sievert teaches the invention as claimed, including the software component of claim 18, where the thread activation component associates the context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event (col. 5 line 59 - col. 6 line 54).

43. As per claim 24, Sievert teaches the invention as claimed, including the method of claim 23, further comprising:

selectively deactivating at least one of the N threads (col. 5 lines 26-38); and

activating at least another of the N threads based on an event using at least one thread activation component (col. 3 lines 45-52).

Art Unit: 2195

44. As per claim 25, Sievert teaches the invention as claimed, including the method of claim 24, where the at least one thread activation component is a completion port (col. 3 lines 20-32).

45. As per claim 26, Sievert teaches the invention as claimed, including the method of claim 24, where selectively deactivating at least one of the N threads comprises deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending (col. 5 lines 26-38).

46. As per claim 27, Sievert teaches the invention as claimed, including the method of claim 26, where activating at least another of the N threads based on an event comprises:

receiving a completion packet using the thread activation component (col. 3 lines 27-32);

and

activating one of the N threads upon receipt of the completion packet using the thread activation component (col. 3 lines 45-52).

47. As per claim 28, Sievert teaches the invention as claimed, including the method of claim 27, where the at least one thread activation component is a completion port (col. 3 lines 20-32).

48. As per claim 33, Sievert teaches the invention as claimed, including the method of claim 23, further comprising:

associating at least one key with at least one of the M requests (col. 5 line 59 - col. 6 line 54);

associating a first one of the N threads with the at least one of the M requests (col. 5 line 59 - col. 6 line 54); and

associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads (col. 5 lines 26-38; col. 5 line 59 - col. 6 line 54).

49. As per claim 34, Sievert teaches the invention as claimed, including the method of claim 33, further comprising associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event (col. 5 line 59 - col. 6 line 54).

50. As per claim 36, Sievert teaches the invention as claimed, including the computer-readable medium of claim 35, further comprising computer-executable instructions for:

selectively deactivating at least one of the N threads (col. 5 lines 26-38); and

activating at least another of the N threads based on an event using at least one thread activation component (col. 3 lines 45-52).

51. As per claim 37, Sievert teaches the invention as claimed, including the computer-readable medium of claim 36, where the at least one thread activation component is a completion port (col. 3 lines 20-32).

52. As per claim 38, Sievert teaches the invention as claimed, including the computer-readable medium of claim 36, where the computer-executable instructions for selectively deactivating at least one of the N threads comprises computer-executable instructions for deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending (col. 5 lines 26-38).

53. As per claim 39, Sievert teaches the invention as claimed, including the computer-readable medium of claim 38, where the computer-executable instructions for activating at least another of the N threads based on an event comprises computer-executable instructions for:

receiving a completion packet using the thread activation component (col. 3 lines 27-32);

and

activating one of the N threads upon receipt of the completion packet using the thread activation component (col. 3 lines 45-52).

54. As per claim 44, Sievert teaches the invention as claimed, including the computer-readable medium of claim 35, further comprising computer-executable instructions for:

associating at least one key with the at least one of the M requests (col. 5 line 59 - col. 6 line 54);

associating a first one of the N threads with the at least one of the M requests (col. 5 line 59 - col. 6 line 54); and

associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads (col. 5 line 59 - col. 6 line 54).

55. As per claim 45, Sievert teaches the invention as claimed, including the computer-readable medium of claim 44, further comprising computer-executable instructions for associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event (col. 5 line 59 - col. 6 line 54).

56. As per claim 47, Sievert teaches the invention as claimed, including the software component of claim 46, further comprising:

means for selectively deactivating at least one of the N threads (col. 5 lines 26-38); and
means for activating at least another of the N threads based on an event (col. 3 lines 45-52).

57. **Claims 14, 29, 40, and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert as applied to claims 13, 28, 39, and 47 above respectively, and further in view of Jones.**

58. As per claim 14, Jones teaches the invention as claimed, including the software component of claim 13, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

59. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

60. As per claim 29, Jones teaches the invention as claimed, including the method of claim 28, further comprising activating an object scheduled to begin sending requests at a specific time using a scheduler thread (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

61. As per claim 40, Jones teaches the invention as claimed, including the computer-readable medium of claim 39, further comprising computer-executable instructions for activating an object scheduled to begin sending requests at a specific time using a scheduler thread (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

62. As per claim 48, Jones teaches the invention as claimed, including the software component of claim 47, further comprising means for activating an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

63. **Claims 15, 30, 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert in view of Jones as applied to claims 14, 29, and 40 above respectively, and further in view of Okano.**

64. As per claim 15, Okano teaches the invention as claimed, including the software component of claim 14, further comprising a DNS thread that resolves domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

65. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, Jones, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

66. As per claim 30, Okano teaches the invention as claimed, including the method of claim 29, further comprising resolving domain names into IP addresses using a DNS thread (col. 12 line 37 - col. 13 line 5).

67. As per claim 41, Okano teaches the invention as claimed, including the computer-readable medium of claim 40, further comprising computer-executable instructions for resolving domain names into IP addresses using a DNS thread (col. 12 line 37 - col. 13 line 5).

Art Unit: 2195

68. **Claims 16, 31, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert in view of Jones in view of Okano as applied to claims 15, 30, and 41 above respectively, and further in view of Paxhia.**

69. As per claim 16, Paxhia teaches the invention as claimed, including the software component of claim 15, further comprising a timeout thread with a list of active sockets and timers associated with each socket, the timeout thread selectively times-out at least one socket according to at least one timer in the list (col. 41 lines 19-28).

70. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, Jones, Okano, and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

71. As per claim 31, Paxhia teaches the invention as claimed, including the method of claim 30, further comprising selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket (col. 41 lines 19-28).

72. As per claim 42, Paxhia teaches the invention as claimed, including the computer-readable medium of claim 41, further comprising computer-executable instructions for

Art Unit: 2195

selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket (col. 41 lines 19-28).

73. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Jones.

74. As per claim 20, Jones teaches the invention as claimed, including the software component of claim 8, further comprising a scheduler thread that activates an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

75. It would have been obvious to one of ordinary skill in the art to combine IBM and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

76. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Okano.

77. As per claim 21, Okano teaches the invention as claimed, including the software component of claim 8, further comprising a DNS thread that resolves domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

78. It would have been obvious to one of ordinary skill in the art to combine IBM and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

79. **Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Paxhia.**

80. As per claim 22, Paxhia teaches the invention as claimed, including the software component of claim 8, further comprising a timeout thread with a list of active sockets and timers associated with each socket, the timeout thread selectively times-out at least one socket according to at least one timer in the list (col. 41 lines 19-28).

81. It would have been obvious to one of ordinary skill in the art to combine IBM and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

82. Claim 49 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert as applied to claim 47 above, and further in view of Okano.

83. As per claim 49, Okano teaches the invention as claimed, including the software component of claim 47, further comprising means for resolving domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

84. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

85. Claim 50 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert as applied to claim 47 above, and further in view of Paxhia.

86. As per claim 50, Paxhia teaches the invention as claimed, including the software component of claim 47, further comprising means for selectively timing out at least one socket according to at least one timer associated with the at least one socket (col. 41 lines 19-28).

87. It would have been obvious to one of ordinary skill in the art to combine IBM and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the

system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

Response to Arguments

88. **Applicant's arguments filed November 21, 2005 have been fully considered but they are not persuasive.**

89. Applicant argues that the claimed software components are statutory subject matter. Applicant cites the Federal Circuit opinion in *Eolas Techs, Inc. v. Microsoft Corp.*, 399 F.3d 1325 (Fed. Cir. 2005), which states "software code alone qualifies as an invention eligible for patenting under these categories, **at least as processes.**" In other words, a software program alone, when drafted as a process or method, is statutory subject matter. However, it is a settled principle that in order to be statutory subject matter, there must be a "useful, **concrete, and tangible** result."

90. The rejected claims are directed to the software component itself, not the process it carries out, as the preamble is drafted as "A...software component" instead of "A computer-implemented method." Moreover, the claims recite elements of the software component that suggest it more closely resembles a data structure or set of software objects rather than a set of steps executed on a computer. That is, a "completion port object," a "thread pool," and a "state machine" can hardly be characterized as process steps. The case cited by Applicant clearly indicates that software components are statutory when they are described as processes and the steps they perform on a computer are described. If the components of the software component

Art Unit: 2195

are the subject matter of the claim, they must be fixed to some tangible form in order to fit into one of the categories of invention including a “machine” or “manufacture.”

91. The remainder of Applicant’s arguments allege that neither Sievert nor IBM teach “a client side state machine associated with the at least one request.” The independent claims have been amended such that all independent claims now recite this feature. The entirety of the arguments rest on the independent claims deficiencies with respect to this feature, or by virtue of the dependent claims failure to remedy the alleged deficiencies of their parent claims. Applicant claims that the *“clear difference between the invention as claimed and the cited document lies in the fact that applicant’s claimed invention associates a state machine to all client requests received from a client application, whereas the cited document merely associates a solitary state machine to the work queue such that the work queue can be in one of the three states enumerated above.”*

92. Applicant’s arguments are not persuasive for two reasons. First, Applicant seems to indicate in the arguments that the claimed state machine requires a unique state machine for each request. However, this interpretation is not required by the claims, which are only limited to “a client side state machine associated with the at least one request.” It is entirely reasonable to read this claim as allowing a single state machine to be associated with every request. The work queue of Sievert, by progressing through various states to process each request, is thereby a state machine associated with every request in the system. Second, giving the claims their broadest reasonable interpretation, there is no need to even show why the work queue of Sievert corresponds to the claimed state machine. That is, it is well known that a computer system is

often considered a sequential state machine (Armstrong et al., USPN 6,957,435, demonstrates that this was within the knowledge of those having ordinary skill in the art around the time of Applicant's invention. See column 5 line 54 - column 6 line 5 of Armstrong, which states: "As is well known, a computer system is a sequential state machine which performs processes."). Once it is accepted that a computer is a state machine, the fact that the computer processes requests clearly meets the claim limitation of "a client side state machine associated with the at least one request." This is true regarding both Sievert and IBM.

Conclusion

93. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J. Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

Art Unit: 2195

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T. An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Syed Ali
January 12, 2006



MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY DE.